



Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Obsolete
Core Processor	CPU32
Core Size	32-Bit Single-Core
Speed	16MHz
Connectivity	EBI/EMI, SCI, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	15
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	132-BQFP Bumpered
Supplier Device Package	132-PQFP (24.13x24.13)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68332gcfc16

1.1 Features

- Central Processing Unit (CPU32)
 - 32-Bit Architecture
 - Virtual Memory Implementation
 - Table Lookup and Interpolate Instruction
 - Improved Exception Handling for Controller Applications
 - High-Level Language Support
 - Background Debugging Mode
 - Fully Static Operation
- System Integration Module (SIM)
 - External Bus Support
 - Programmable Chip-Select Outputs
 - System Protection Logic
 - Watchdog Timer, Clock Monitor, and Bus Monitor
 - Two 8-Bit Dual Function Input/Output Ports
 - One 7-Bit Dual Function Output Port
 - Phase-Locked Loop (PLL) Clock System
- Time Processor Unit (TPU)
 - Dedicated Microengine Operating Independently of CPU32
 - 16 Independent, Programmable Channels and Pins
 - Any Channel can Perform any Time Function
 - Two Timer Count Registers with Programmable Prescalers
 - Selectable Channel Priority Levels
- Queued Serial Module (QSM)
 - Enhanced Serial Communication Interface
 - Queued Serial Peripheral Interface
 - One 8-Bit Dual Function Port
- Static RAM Module with TPU Emulation Capability (TPURAM)
 - 2-Kbytes of Static RAM
 - May be Used as Normal RAM or TPU Microcode Emulation RAM

1.2 Block Diagram

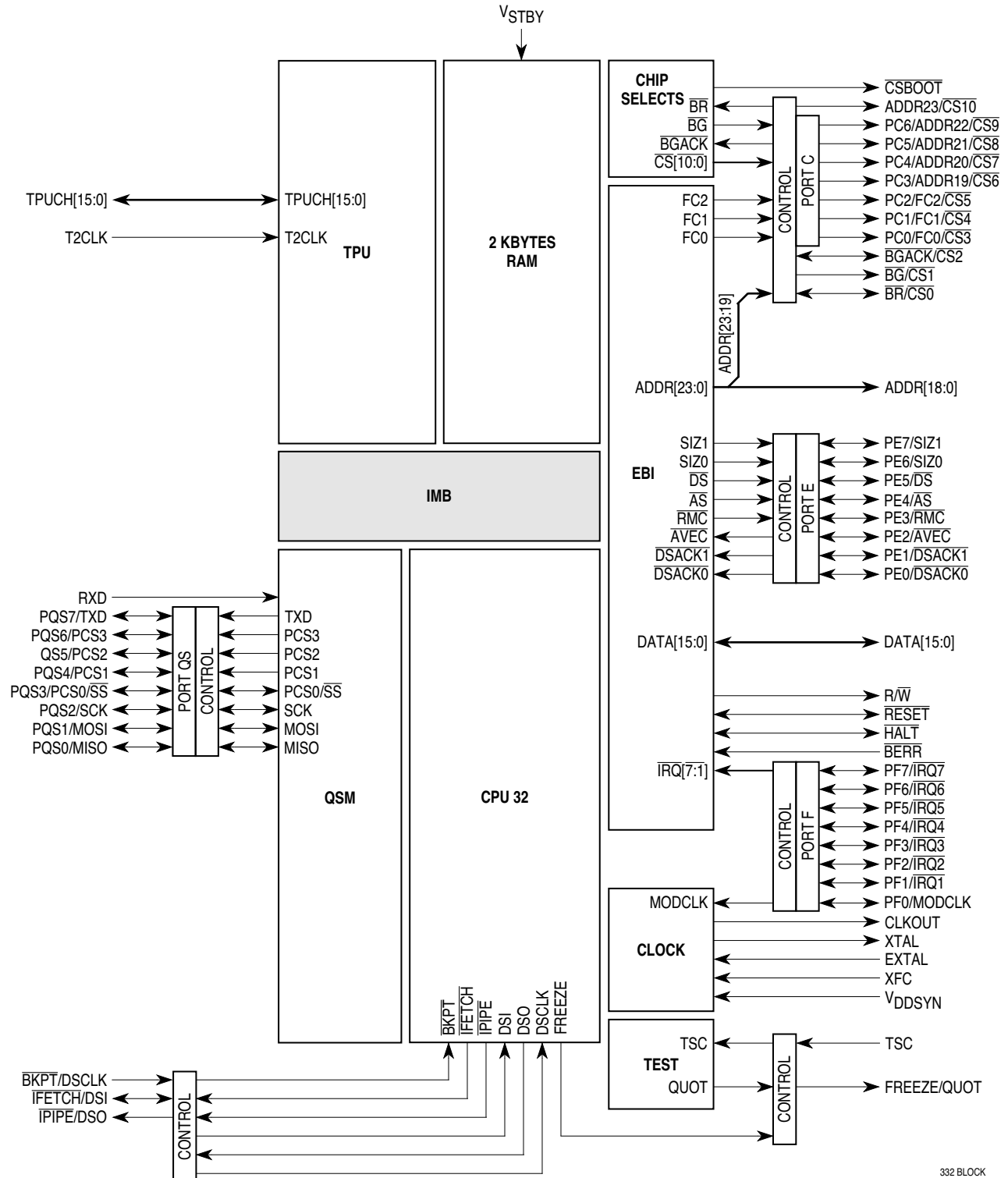


Figure 1 MCU Block Diagram

2 Signal Descriptions

2.1 Pin Characteristics

The following table shows MCU pins and their characteristics. All inputs detect CMOS logic levels. All inputs can be put in a high-impedance state, but the method of doing this differs depending upon pin function. Refer to the table, MCU Driver Types, for a description of output drivers. An entry in the discrete I/O column of the MCU Pin Characteristics table indicates that a pin has an alternate I/O function. The port designation is given when it applies. Refer to the MCU Block Diagram for information about port organization.

Table 2 MCU Pin Characteristic

Pin Mnemonic	Output Driver	Input Synchronized	Input Hysteresis	Discrete I/O	Port Designation
ADDR23/CS10/ECLK	A	Y	N	O	—
ADDR[22:19]/CS[9:6]	A	Y	N	O	PC[6:3]
ADDR[18:0]	A	Y	N	—	—
\overline{AS}	B	Y	N	I/O	PE5
\overline{AVEC}	B	Y	N	I/O	PE2
\overline{BERR}	B	Y	N	—	—
$\overline{BG/CS1}$	B	—	—	—	—
$\overline{BGACK/CS2}$	B	Y	N	—	—
$\overline{BKPT/DSCLK}$	—	Y	Y	—	—
$\overline{BR/CS0}$	B	Y	N	—	—
CLKOUT	A	—	—	—	—
\overline{CSBOOT}	B	—	—	—	—
DATA[15:0] ¹	Aw	Y	N	—	—
\overline{DS}	B	Y	N	I/O	PE4
$\overline{DSACK1}$	B	Y	N	I/O	PE1
$\overline{DSACK0}$	B	Y	N	I/O	PE0
DSI/IFETCH	A	Y	Y	—	—
DSO/IPIPE	A	—	—	—	—
EXTAL ²	—	—	Special	—	—
FC[2:0]/CS[5:3]	A	Y	N	O	PC[2:0]
FREEZE/QUOT	A	—	—	—	—
\overline{HALT}	Bo	Y	N	—	—
$\overline{IRQ[7:1]}$	B	Y	Y	I/O	PF[7:1]
MISO	Bo	Y	Y	I/O	PQS0
MODCLK ¹	B	Y	N	I/O	PF0
MOSI	Bo	Y	Y	I/O	PQS1
PCS0/ \overline{SS}	Bo	Y	Y	I/O	PQS3
PCS[3:1]	Bo	Y	Y	I/O	PQS[6:4]
R/W	A	Y	N	—	—
\overline{RESET}	Bo	Y	Y	—	—
\overline{RMC}	B	Y	N	I/O	PE3
RXD	—	N	N	—	—
SCK	Bo	Y	Y	I/O	PQS2
SIZ[1:0]	B	Y	N	I/O	PE[7:6]

3.2.7 Periodic Interrupt Timer

The periodic interrupt timer (PIT) generates interrupts of specified priorities at specified intervals. Timing for the PIT is provided by a programmable prescaler driven by the system clock.

PICR — Periodic Interrupt Control Register

\$YFFA22

15	14	13	12	11	10	8	7	0
0	0	0	0	0	PIRQL		PIV	

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

This register contains information concerning periodic interrupt priority and vectoring. Bits [10:0] can be read or written at any time. Bits [15:11] are unimplemented and always return zero.

PIRQL[2:0] — Periodic Interrupt Request Level

The following table shows what interrupt request level is asserted when a periodic interrupt is generated. If a PIT interrupt and an external \overline{IRQ} signal of the same priority occur simultaneously, the PIT interrupt is serviced first. The periodic timer continues to run when the interrupt is disabled.

PIRQL	Interrupt Request Level
000	Periodic Interrupt Disabled
001	Interrupt Request Level 1
010	Interrupt Request Level 2
011	Interrupt Request Level 3
100	Interrupt Request Level 4
101	Interrupt Request Level 5
110	Interrupt Request Level 6
111	Interrupt Request Level 7

PIV[7:0] — Periodic Interrupt Vector

The bits of this field contain the vector generated in response to an interrupt from the periodic timer. When the SIM responds, the periodic interrupt vector is placed on the bus.

PITR — Periodic Interrupt Timer Register

\$YFFA24

15	14	13	12	11	10	9	8	7	0
0	0	0	0	0	0	0	PTP	PITM	

RESET:

0 0 0 0 0 0 0 MODCLK 0 0 0 0 0 0 0 0

The PITR contains the count value for the periodic timer. A zero value turns off the periodic timer. This register can be read or written at any time.

PTP — Periodic Timer Prescaler Control

0 = Periodic timer clock not prescaled

1 = Periodic timer clock prescaled by a value of 512

The reset state of PTP is the complement of the state of the MODCLK signal during reset.

PITM[7:0] — Periodic Interrupt Timing Modulus Field

This is an 8-bit timing modulus. The period of the timer can be calculated as follows:

$$\text{PIT Period} = [(PITM)(\text{Prescaler})(4)]/\text{EXTAL}$$

where

PIT Period = Periodic interrupt timer period

PITM = Periodic interrupt timer register modulus (PITR[7:0])

EXTAL Frequency = Crystal frequency

Prescale = 512 or 1 depending on the state of the PTP bit in the PITR

When an external system clock signal is applied (i.e., the PLL is not used), duty cycle of the input is critical, especially at near maximum operating frequencies. The relationship between clock signal duty cycle and clock signal period is expressed:

$$\text{Minimum external clock period} = \frac{\text{minimum external clock high/low time}}{50\% - \text{percentage variation of external clock input duty cycle}}$$

3.3.2 Clock Synthesizer Operation

A voltage controlled oscillator (VCO) generates the system clock signal. A portion of the clock signal is fed back to a divider/counter. The divider controls the frequency of one input to a phase comparator. The other phase comparator input is a reference signal, either from the internal oscillator or from an external source. The comparator generates a control signal proportional to the difference in phase between its two inputs. The signal is low-pass filtered and used to correct VCO output frequency.

The synthesizer locks when VCO frequency is identical to reference frequency. Lock time is affected by the filter time constant and by the amount of difference between the two comparator inputs. Whenever comparator input changes, the synthesizer must re-lock. Lock status is shown by the SLOCK bit in SYNCR.

The MCU does not come out of reset state until the synthesizer locks. Crystal type, characteristic frequency, and layout of external oscillator circuitry affect lock time.

The low-pass filter requires an external low-leakage capacitor, typically 0.1 μF , connected between the XFC and V_{DDSYN} pins.

V_{DDSYN} is used to power the clock circuits. A separate power source increases MCU noise immunity and can be used to run the clock when the MCU is powered down. Use a quiet power supply as the V_{DDSYN} source, since PLL stability depends on the VCO, which uses this supply. Place adequate external bypass capacitors as close as possible to the V_{DDSYN} pin to ensure stable operating frequency.

When the clock synthesizer is used, control register SYNCR determines operating frequency and various modes of operation. SYNCR can be read only when the processor is operating at the supervisor privilege level.

The SYNCR X bit controls a divide by two prescaler that is not in the synthesizer feedback loop. Setting X doubles clock speed without changing VCO speed. There is no VCO relock delay. The SYNCR W bit controls a 3-bit prescaler in the feedback divider. Setting W increases VCO speed by a factor of four. The SYNCR Y field determines the count modulus for a modulo 64 down counter, causing it to divide by a value of $Y + 1$. When either W or Y value changes, there is a VCO relock delay.

Clock frequency is determined by SYNCR bit settings as follows:

$$F_{\text{SYSTEM}} = F_{\text{REFERENCE}} [4(Y + 1)(2^{2W} + X)]$$

In order for the device to perform correctly, the clock frequency selected by the W, X, and Y bits must be within the limits specified for the MCU.

The VCO frequency is twice the system clock frequency if $X = 1$ or four times the system clock frequency if $X = 0$.

The reset state of SYNCR (\$3F00) produces a modulus-64 count.

3.4 External Bus Interface

The external bus interface (EBI) transfers information between the internal MCU bus and external devices. The external bus has 24 address lines and 16 data lines.

The EBI provides dynamic sizing between 8-bit and 16-bit data accesses. It supports byte, word, and long-word transfers. Ports are accessed through the use of asynchronous cycles controlled by the data transfer (SIZ1 and SIZ0) and data size acknowledge pins (DSACK1 and DSACK0). Multiple bus cycles may be required for a transfer to or from an 8-bit port.

Port width is the maximum number of bits accepted or provided during a bus transfer. External devices must follow the handshake protocol described below. Control signals indicate the beginning of the cycle, the address space, the size of the transfer, and the type of cycle. The selected device controls the length of the cycle. Strobe signals, one for the address bus and another for the data bus, indicate the validity of an address and provide timing information for data. The EBI operates in an asynchronous mode for any port width.

To add flexibility and minimize the necessity for external logic, MCU chip-select logic can be synchronized with EBI transfers. Chip-select logic can also provide internally-generated bus control signals for these accesses. Refer to **3.5 Chip Selects** for more information.

3.4.1 Bus Control Signals

The CPU initiates a bus cycle by driving the address, size, function code, and read/write outputs. At the beginning of the cycle, size signals SIZ0 and SIZ1 are driven along with the function code signals. The size signals indicate the number of bytes remaining to be transferred during an operand cycle. They are valid while the address strobe (\overline{AS}) is asserted. The following table shows SIZ0 and SIZ1 encoding. The read/write (R/\overline{W}) signal determines the direction of the transfer during a bus cycle. This signal changes state, when required, at the beginning of a bus cycle, and is valid while \overline{AS} is asserted. R/\overline{W} only changes state when a write cycle is preceded by a read cycle or vice versa. The signal can remain low for two consecutive write cycles.

Table 8 Size Signal Encoding

SIZ1	SIZ0	Transfer Size
0	1	Byte
1	0	Word
1	1	Three Byte
0	0	Long Word

3.4.2 Function Codes

The CPU32 automatically generates function code signals FC[2:0]. The function codes can be considered address extensions that automatically select one of eight address spaces to which an address applies. These spaces are designated as either user or supervisor, and program or data spaces. Address space 7 is designated CPU space. CPU space is used for control information not normally associated with read or write bus cycles. Function codes are valid while \overline{AS} is asserted.

ADDR0 also affects the operation of the data multiplexer. During an operand transfer, ADDR[23:1] indicate the word base address of the portion of the operand to be accessed, and ADDR0 indicates the byte offset from the base.

3.4.11 Misaligned Operands

CPU32 processor architecture uses a basic operand size of 16 bits. An operand is misaligned when it overlaps a word boundary. This is determined by the value of ADDR0. When ADDR0 = 0 (an even address), the address is on a word and byte boundary. When ADDR0 = 1 (an odd address), the address is on a byte boundary only. A byte operand is aligned at any address; a word or long-word operand is misaligned at an odd address. The CPU32 does not support misaligned operand transfers.

The largest amount of data that can be transferred by a single bus cycle is an aligned word. If the MCU transfers a long-word operand via a 16-bit port, the most significant operand word is transferred on the first bus cycle and the least significant operand word on a following bus cycle.

3.4.12 Operand Transfer Cases

The following table summarizes how operands are aligned for various types of transfers. OPn entries are portions of a requested operand that are read or written during a bus cycle and are defined by SIZ1, SIZ0, and ADDR0 for that bus cycle.

Table 11 Operand Alignment

Transfer Case	SIZ1	SIZ0	ADDR0	DSACK1	DSACK0	DATA [15:8]	DATA [7:0]
Byte to 8-Bit Port (Even/Odd)	0	1	X	1	0	OP0	(OP0)
Byte to 16-Bit Port (Even)	0	1	0	0	X	OP0	(OP0)
Byte to 16-Bit Port (Odd)	0	1	1	0	X	(OP0)	OP0
Word to 8-Bit Port (Aligned)	1	0	0	1	0	OP0	(OP1)
Word to 8-Bit Port (Misaligned) ³	1	0	1	1	0	OP0	(OP0)
Word to 16-Bit Port (Aligned)	1	0	0	0	X	OP0	OP1
Word to 16-Bit Port (Misaligned) ³	1	0	1	0	X	(OP0)	OP0
3 Byte to 8-Bit Port (Aligned) ²	1	1	0	1	0	OP0	(OP1)
3 Byte to 8-Bit Port (Misaligned) ^{2, 3}	1	1	1	1	0	OP0	(OP0)
3 Byte to 16-Bit Port (Aligned) ²	1	1	0	0	X	OP0	OP1
3 Byte to 16-Bit Port (Misaligned) ^{2, 3}	1	1	1	0	X	(OP0)	OP0
Long Word to 8-Bit Port (Aligned)	0	0	0	1	0	OP0	(OP1)
Long Word to 8-Bit Port (Misaligned) ³	1	0	1	1	0	OP0	(OP0)
Long Word to 16-Bit Port (Aligned)	0	0	0	0	X	OP0	OP1
Long Word to 16-Bit Port (Misaligned) ³	1	0	1	0	X	(OP0)	OP0

NOTES:

1. Operands in parentheses are ignored by the CPU32 during read cycles.
2. Three-byte transfer cases occur only as a result of a long word to byte transfer.
3. The CPU32 does not support misaligned word or long-word transfers.

3.5 Chip Selects

Typical microcontrollers require additional hardware to provide external chip-select signals. Twelve independently programmable chip selects provide fast two-cycle access to external memory or peripherals. Address block sizes of 2 Kbytes to 1 Mbyte can be selected.

CSORBT —Chip-Select Option Register Boot ROM

\$YFFA4A

15	14	13	12	11	10	9	6	5	4	3	1	0
MODE	BYTE	R/W	STRB	DSACK	SPACE	IPL	AVEC					

RESET:

0 1 1 1 1 0 1 1 0 1 1 0 0 0

CSOR[10:0] —Chip-Select Option Registers

\$YFFA4E–\$YFFA76

15	14	13	12	11	10	9	6	5	4	3	1	0
MODE	BYTE	R/W	STRB	DSACK	SPACE	IPL	AVEC					

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0

CSORBT, the option register for \overline{CSBOOT} , contains special reset values that support bootstrap operations from peripheral memory devices.

The following bit descriptions apply to both CSORBT and CSOR[10:0] option registers.

MODE — Asynchronous/Synchronous Mode

0 = Asynchronous mode selected (chip-select assertion determined by internal or external bus control signals)

1 = Synchronous mode selected (chip-select assertion synchronized with ECLK signal)

In asynchronous mode, the chip select is asserted synchronized with \overline{AS} or \overline{DS} .

The \overline{DSACK} field is not used in synchronous mode because a bus cycle is only performed as a synchronous operation. When a match condition occurs on a chip select programmed for synchronous operation, the chip select signals the EBI that an ECLK cycle is pending.

BYTE — Upper/Lower Byte Option

This field is used only when the chip-select 16-bit port option is selected in the pin assignment register. The following table lists upper/lower byte options.

Byte	Description
00	Disable
01	Lower Byte
10	Upper Byte
11	Both Bytes

R/W — Read/Write

This field causes a chip select to be asserted only for a read, only for a write, or for both read and write. Refer to the following table for options available.

R/W	Description
00	Reserved
01	Read Only
10	Write Only
11	Read/Write

STRB — Address Strobe/Data Strobe

0 = Address strobe

1 = Data strobe

This bit controls the timing for assertion of a chip select in asynchronous mode. Selecting address strobe causes chip select to be asserted synchronized with address strobe. Selecting data strobe causes chip select to be asserted synchronized with data strobe.

PEPAR — Port E Pin Assignment Register

\$YFFA17

15	8	7	6	5	4	3	2	1	0
NOT USED								PEPA7	PEPA6

RESET:

DATA8 DATA8 DATA8 DATA8 DATA8 DATA8 DATA8 DATA8

The bits in this register control the function of each port E pin. Any bit set to one configures the corresponding pin as a bus control signal, with the function shown in the following table. Any bit cleared to zero defines the corresponding pin to be an I/O pin, controlled by PORTE and DDRE.

Data bus bit 8 controls the state of this register following reset. If DATA8 is set to one during reset, the register is set to \$FF, which defines all port E pins as bus control signals. If DATA8 is cleared to zero during reset, this register is set to \$00, configuring all port E pins as I/O pins.

Any bit cleared to zero defines the corresponding pin to be an I/O pin. Any bit set to one defines the corresponding pin to be a bus control signal.

Table 16 Port E Pin Assignments

PEPAR Bit	Port E Signal	Bus Control Signal
PEPA7	PE7	SIZ1
PEPA6	PE6	SIZ0
PEPA5	PE5	\overline{AS}
PEPA4	PE4	\overline{DS}
PEPA3	PE3	\overline{RMC}
PEPA2	PE2	\overline{AVEC}
PEPA1	PE1	$\overline{DSACK1}$
PEPA0	PE0	$\overline{DSACK0}$

PORTF0, PORTF1 — Port F Data Register

\$YFFA19, \$YFFA1B

15	8	7	6	5	4	3	2	1	0
NOT USED								PF7	PF6

RESET:

U U U U U U U U

The write to the port F data register is stored in the internal data latch, and if any port F pin is configured as an output, the value stored for that bit is driven onto the pin. A read of the port F data register returns the value at the pin only if the pin is configured as a discrete input. Otherwise, the value read is the value stored in the register.

The port F data register is a single register that can be accessed in two locations. When accessed at \$YFFA19, the register is referred to as PORTF0; when accessed at \$YFFA1B, the register is referred to as PORTF1. The register can be read or written at any time. It is unaffected by reset.

DDRF — Port F Data Direction Register

\$YFFA1D

15	8	7	6	5	4	3	2	1	0
NOT USED								DDF7	DDF6

RESET:

0 0 0 0 0 0 0 0

The bits in this register control the direction of the pin drivers when the pins are configured for I/O. Any bit in this register set to one configures the corresponding pin as an output. Any bit in this register cleared to zero configures the corresponding pin as an input.

Table 18 Reset Mode Selection

DATA0	CSBOOT 16-Bit	CSBOOT 8-Bit
DATA1	CS0 CS1 CS2	BR BG BGACK
DATA2	CS3 CS4 CS5	FC0 FC1 FC2
DATA3 DATA4 DATA5 DATA6 DATA7	CS6 CS[7:6] CS[8:6] CS[9:6] CS[10:6]	ADDR19 ADDR[20:19] ADDR[21:19] ADDR[22:19] ADDR[23:19]
DATA8	DSACK0, DSACK1, AVEC, DS, AS, SIZ[1:0]	PORTE
DATA9	IRQ[7:1] MODCLK	PORTF
DATA11	Test Mode Disabled	Test Mode Enabled
MODCLK	VCO = System Clock	EXTAL = System Clock
BKPT	Background Mode Disabled	Background Mode Enabled

3.7.2 Functions of Pins for Other Modules During Reset

Generally, pins associated with modules other than the SIM default to port functions, and input/output ports are set to input state. This is accomplished by disabling pin functions in the appropriate control registers, and by clearing the appropriate port data direction registers. Refer to individual module sections in this manual for more information. The following table is a summary of module pin function out of reset.

Table 19 Module Pin Functions

Module	Pin Mnemonic	Function
CPU32	DSI/IFETCH	DSI/IFETCH
	DSO/IPIPE	DSO/IPIPE
	BKPT/DSCLK	BKPT/DSCLK
GPT	PGP7/IC4/OC5	Discrete Input
	PGP[6:3]/OC[4:1]	Discrete Input
	PGP[2:0]/IC[3:1]	Discrete Input
	PAI	Discrete Input
	PCLK	Discrete Input
	PWMA, PWMB	Discrete Output
QSM	PQS7/TXD	Discrete Input
	PQS[6:4]/PCS[3:1]	Discrete Input
	PQS3/PCS0/SS	Discrete Input
	PQS2/SCK	Discrete Input
	PQS1/MOSI	Discrete Input
	PQS0/MISO	Discrete Input
	RXD	RXD

1. The dominant interrupt source supplies a vector number and \overline{DSACK} signals appropriate to the access. The CPU32 acquires the vector number.
 2. The \overline{AVEC} signal is asserted (the signal can be asserted by the dominant interrupt source or the pin can be tied low), and the CPU32 generates an autovector number corresponding to interrupt priority.
 3. The bus monitor asserts \overline{BERR} and the CPU32 generates the spurious interrupt vector number.
- F. The vector number is converted to a vector address.
- G. The content of the vector address is loaded into the PC, and the processor transfers control to the exception handler routine.

3.9 Factory Test Block

The test submodule supports scan-based testing of the various MCU modules. It is integrated into the SIM to support production testing.

Test submodule registers are intended for Motorola use. Register names and addresses are provided to indicate that these addresses are occupied.

SIMTR —System Integration Test Register	\$YFFA02
SIMTRE —System Integration Test Register (E Clock)	\$YFFA08
TSTMSRA —Master Shift Register A	\$YFFA30
TSTMSRB —Master Shift Register B	\$YFFA32
TSTSC —Test Module Shift Count	\$YFFA34
TSTRC —Test Module Repetition Count	\$YFFA36
CREG —Test Module Control Register	\$YFFA38
DREG —Test Module Distributed Register	\$YFFA3A

4.3 Status Register

The status register contains the condition codes that reflect the results of a previous operation and can be used for conditional instruction execution in a program. The lower byte containing the condition codes is the only portion of the register available at the user privilege level; it is referenced as the condition code register (CCR) in user programs. At the supervisor privilege level, software can access the full status register, including the interrupt priority mask and additional control bits.

SR —Status Register

15	14	13	12	11	10	8	7	6	5	4	3	2	1	0
T1	T0	S	0	0	IP	0	0	0	X	N	Z	V	C	
RESET:														
0	0	1	0	0	1	1	1	0	0	0	U	U	U	U

System Byte

- T[1:0] —Trace Enable
- S —Supervisor/User State
- Bits [12:11] —Unimplemented
- IP[2:0] —Interrupt Priority Mask

User Byte (Condition Code Register)

- Bits [7:5] — Unimplemented
- X — Extend
- N — Negative
- Z — Zero
- V — Overflow
- C — Carry

4.4 Data Types

Six basic data types are supported:

- Bits
- Packed Binary Coded Decimal Digits
- Byte Integers (8 bits)
- Word Integers (16 bits)
- Long-Word Integers (32 bits)
- Quad-Word Integers (64 bits)

4.5 Addressing Modes

Addressing in the CPU32 is register-oriented. Most instructions allow the results of the specified operation to be placed either in a register or directly in memory. This flexibility eliminates the need for extra instructions to store register contents in memory. The CPU32 supports seven basic addressing modes:

- Register direct
- Register indirect
- Register indirect with index
- Program counter indirect with displacement
- Program counter indirect with index
- Absolute
- Immediate

Included in the register indirect addressing modes are the capabilities to post-increment, predecrement, and offset. The program counter relative mode also has index and offset capabilities. In addition to these addressing modes, many instructions implicitly specify the use of the status register, stack pointer, or program counter.

5.5.2 Channel Control Registers

CIER — Channel Interrupt Enable Register

\$YFFE0A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 15	CH 14	CH 13	CH 12	CH 11	CH 10	CH 9	CH 8	CH 7	CH 6	CH 5	CH 4	CH 3	CH 2	CH 1	CH 0

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

CH[15:0] — Channel Interrupt Enable/Disable

0 = Channel interrupts disabled

1 = Channel interrupts enabled

CISR — Channel Interrupt Status Register

\$YFFE20

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 15	CH 14	CH 13	CH 12	CH 11	CH 10	CH 9	CH 8	CH 7	CH 6	CH 5	CH 4	CH 3	CH 2	CH 1	CH 0

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

CH[15:0] — Channel Interrupt Status Bit

0 = Channel interrupt not asserted

1 = Channel interrupt asserted

CFSR0 — Channel Function Select Register 0

\$YFFE0C

15	12	11	8	7	4	3	0
CHANNEL15	CHANNEL14	CHANNEL13	CHANNEL12	CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

CFSR1 — Channel Function Select Register 1

\$YFFE0E

15	12	11	8	7	4	3	0
CHANNEL11	CHANNEL10	CHANNEL9	CHANNEL8	CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

CFSR2 — Channel Function Select Register 2

\$YFFE10

15	12	11	8	7	4	3	0
CHANNEL7	CHANNEL6	CHANNEL5	CHANNEL4	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

CFSR3 — Channel Function Select Register 3

\$YFFE12

15	12	11	8	7	4	3	0
CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0	CHANNEL15	CHANNEL14	CHANNEL13	CHANNEL12

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

CHANNEL[15:0] — Encoded Time Function for each Channel

Encoded 4-bit fields in the channel function select registers specify one of 16 time functions to be executed on the corresponding channel.

6.3 Pin Function

The following table is a summary of the functions of the QSM pins when they are not configured for general-purpose I/O. The QSM data direction register (DDRQS) designates each pin except RXD as an input or output.

	Pin	Mode	Pin Function
QSPI Pins	MISO	Master	Serial Data Input to QSPI
		Slave	Serial Data Output from QSPI
	MOSI	Master	Serial Data Output from QSPI
		Slave	Serial Data Input to QSPI
	SCK	Master	Clock Output from QSPI
		Slave	Clock Input to QSPI
SCI Pins	PCS0/SS	Master	Input: Assertion Causes Mode Fault Output: Selects Peripherals
		Slave	Input: Selects the QSPI
	PCS[3:1]	Master	Output: Selects Peripherals
		Slave	None
	TXD	Transmit	Serial Data Output from SCI
	RXD	Receive	Serial Data Input to SCI

6.4 QSM Registers

QSM registers are divided into four categories: QSM global registers, QSM pin control registers, QSPI submodule registers, and SCI submodule registers. The QSPI and SCI registers are defined in separate sections below. Writes to unimplemented register bits have no meaning or effect, and reads from unimplemented bits always return a logic zero value.

The module mapping bit of the SIM configuration register (SIMCR) defines the most significant bit (ADDR23) of the address, shown in each register figure as Y (Y = \$7 or \$F). This bit, concatenated with the rest of the address given, forms the absolute address of each register. Refer to the SIM section of this technical summary for more information about how the state of MM affects the system.

6.4.1 Global Registers

The QSM global registers contain system parameters used by both the QSPI and the SCI submodules. These registers contain the bits and fields used to configure the QSM.

QSMCR — QSM Configuration Register

\$YFFC00

15	14	13	12	11	10	9	8	7	6	5	4	3	0
STOP	FRZ1	FRZ0	0	0	0	0	0	SUPV	0	0	0	IARB	

RESET:

0 0 0 0 0 0 0 0 1 0 0 0 0 0

The QSMCR contains parameters for the QSM/CPU/intermodule bus (IMB) interface.

STOP — Stop Enable

0 = Normal QSM clock operation

1 = QSM clock operation stopped

STOP places the QSM in a low-power state by disabling the system clock in most parts of the module. The QSMCR is the only register guaranteed to be readable while STOP is asserted. The QSPI RAM is not readable. However, writes to RAM or any register are guaranteed to be valid while STOP is asserted. STOP can be negated by the CPU and by reset.

The system software must stop each submodule before asserting STOP to avoid complications at re-start and to avoid data corruption. The SCI submodule receiver and transmitter should be disabled, and the operation should be verified for completion before asserting STOP. The QSPI submodule should be stopped by asserting the HALT bit in SPCR3 and by asserting STOP after the HALTA flag is set.

FRZ1 — Freeze 1

- 0 = Ignore the FREEZE signal on the IMB
- 1 = Halt the QSPI (on a transfer boundary)

FRZ1 determines what action is taken by the QSPI when the FREEZE signal of the IMB is asserted. FREEZE is asserted whenever the CPU enters the background mode.

FRZ0 — Freeze 0

Reserved

Bits [12:8] — Not Implemented

SUPV — Supervisor/Unrestricted

- 0 = User access
- 1 = Supervisor access

SUPV defines the assignable QSM registers as either supervisor-only data space or unrestricted data space.

IARB — Interrupt Arbitration Identification Number

The IARB field is used to arbitrate between simultaneous interrupt requests of the same priority. Each module that can generate interrupt requests must be assigned a unique, non-zero IARB field value. Refer to **3.8 Interrupts** for more information.

QTEST — QSM Test Register

\$YFFC02

QTEST is used during factory testing of the QSM. Accesses to QTEST must be made while the MCU is in test mode.

QILR — QSM Interrupt Levels Register

\$YFFC04

15	14	13	11	10	8	7	0
0	0	ILQSPI			ILSCI		QIVR

RESET:

0 0 0 0 0 0 0 0

QILR determines the priority level of interrupts requested by the QSM and the vector used when an interrupt is acknowledged.

ILQSPI — Interrupt Level for QSPI

ILQSPI determines the priority of QSPI interrupts. This field must be given a value between \$0 (interrupts disabled) to \$7 (highest priority).

ILSCI — Interrupt Level of SCI

ILSCI determines the priority of SCI interrupts. This field must be given a value between \$0 (interrupts disabled) to \$7 (highest priority).

If ILQSPI and ILSCI are the same nonzero value, and both submodules simultaneously request interrupt service, QSPI has priority.

6.5 QSPI Submodule

The QSPI submodule communicates with external devices through a synchronous serial bus. The QSPI is fully compatible with the serial peripheral interface (SPI) systems found on other Motorola products. A block diagram of the QSPI is shown below.

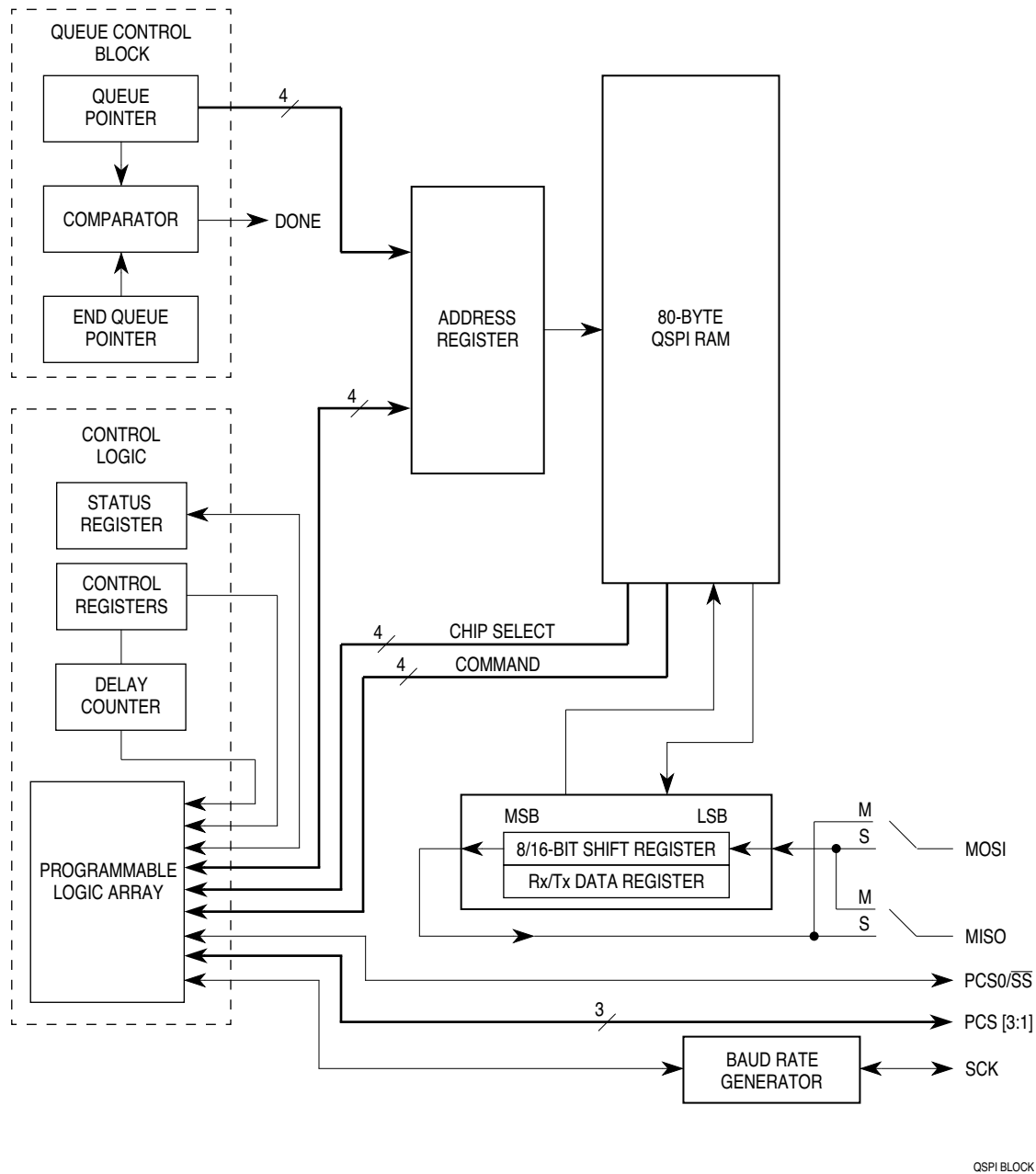


Figure 14 QSPI Block Diagram

6.5.1 QSPI Pins

Seven pins are associated with the QSPI. When not needed for a QSPI application, they can be configured as general-purpose I/O pins. The **PCS0/SS** pin can function as a peripheral chip select output, slave select input, or general-purpose I/O. Refer to the following table for QSPI input and output pins and their functions.

SCK baud rate:

$$\text{SCK Baud Rate} = \text{System Clock}/(2\text{SPBR})$$

or

$$\text{SPBR} = \text{System Clock}/(2\text{SCK})(\text{Baud Rate Desired})$$

where SPBR equals {2, 3, 4,..., 255}

Giving SPBR a value of zero or one disables the baud rate generator. SCK is disabled and assumes its inactive state value. No serial transfers occur. At reset, baud rate is initialized to one eighth of the system clock frequency.

SPCR1 — QSPI Control Register 1

\$YFFC1A

15	14	8	7	0
SPE	DSCKL		DTL	

RESET:

0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0

SPCR1 contains parameters for configuring the QSPI before it is enabled. The CPU can read and write this register, but the QSM has read access only, except for SPE, which is automatically cleared by the QSPI after completing all serial transfers, or when a mode fault occurs.

SPE — QSPI Enable

0 = QSPI is disabled. QSPI pins can be used for general-purpose I/O.

1 = QSPI is enabled. Pins allocated by PQSPAR are controlled by the QSPI.

DSCKL — Delay before SCK

When the DSCK bit in command RAM is set, this field determines the length of delay from PCS valid to SCK transition. PCS can be any of the four peripheral chip-select pins. The following equation determines the actual delay before SCK:

$$\text{PCS to SCK Delay} = [\text{DSCKL}/\text{System Clock}]$$

where DSCKL equals {1, 2, 3,..., 127}.

When the DSCK value of a queue entry equals zero, then DSCKL is not used. Instead, the PCS valid-to-SCK transition is one-half SCK period.

DTL — Length of Delay after Transfer

When the DT bit in command RAM is set, this field determines the length of delay after serial transfer. The following equation is used to calculate the delay:

$$\text{Delay after Transfer} = [(32\text{DTL})/\text{System Clock}]$$

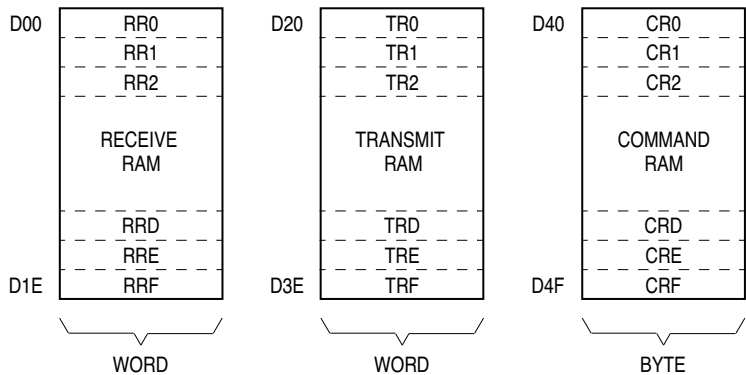
where DTL equals {1, 2, 3,..., 255}.

A zero value for DTL causes a delay-after-transfer value of 8192/System Clock.

If DT equals zero, a standard delay is inserted.

$$\text{Standard Delay after Transfer} = [17/\text{System Clock}]$$

Delay after transfer can be used to provide a peripheral deselect interval. A delay can also be inserted between consecutive transfers to allow serial A/D converters to complete conversion.



QSPI RAM MAP

Figure 15 QSPI RAM

Once the CPU has set up the queue of QSPI commands and enabled the QSPI, the QSPI can operate independently of the CPU. The QSPI executes all of the commands in its queue, sets a flag indicating that it is finished, and then either interrupts the CPU or waits for CPU intervention. It is possible to execute a queue of commands repeatedly without CPU intervention.

RR[0:F] — Receive Data RAM **\$YFFD00**

Data received by the QSPI is stored in this segment. The CPU reads this segment to retrieve data from the QSPI. Data stored in receive RAM is right-justified. Unused bits in a receive queue entry are set to zero by the QSPI upon completion of the individual queue entry. The CPU can access the data using byte, word, or long-word addressing.

The CPTQP value in SPSR shows which queue entries have been executed. The CPU uses this information to determine which locations in receive RAM contain valid data before reading them.

TR[0:F] — Transmit Data RAM **\$YFFD20**

Data that is to be transmitted by the QSPI is stored in this segment. The CPU usually writes one word of data into this segment for each queue command to be executed.

Information to be transmitted must be written to transmit data RAM in a right-justified format. The QSPI cannot modify information in the transmit data RAM. The QSPI copies the information to its data serializer for transmission. Information remains in transmit RAM until overwritten.

CR[0:F] — Command RAM **\$YFFD40**

7	6	5	4	3	2	1	0
CONT	BITSE	DT	DSCK	PCS3	PCS2	PCS1	PCS0*
—	—	—	—	—	—	—	—
CONT	BITSE	DT	DSCK	PCS3	PCS2	PCS1	PCS0*

COMMAND CONTROL

PERIPHERAL CHIP SELECT

*The PCS0 bit represents the dual-function PCS0/ \overline{SS} .

Writing a value of zero to SCBR disables the baud rate generator.

The following table lists the SCBR settings for standard and maximum baud rates using 16.78-MHz and 20.97-MHz system clocks.

Table 27 SCI Baud Rates

Nominal Baud Rate	Actual Rate with 16.78-MHz Clock	SCBR Value	Actual Rate with 20.97-MHz Clock	SCBR Value
64*	64.0	\$1FFF	—	—
110	110.0	\$129E	110.0	\$1745
300	299.9	\$06D4	300.1	\$0888
600	599.9	\$036A	600.1	\$0444
1200	1199.7	\$0165	1200.3	\$0222
2400	2405.0	\$00DA	2400.6	\$0111
4800	4810.0	\$006D	4783.6	\$0089
9600	9532.5	\$0037	9637.6	\$0044
19200	19418.1	\$0016	19275.3	\$0022
38400	37449.1	\$000E	38550.6	\$0011
76800	74898.3	\$0007	72817.8	\$0009
Maximum Rate	524288.0	\$0001	655360.0	\$0001

SCCR1 — SCI Control Register 1

\$YFFC0A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	LOOPS	WOMS	ILT	PT	PE	M	WAKE	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK

RESET:

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

SCCR1 contains SCI configuration parameters. The CPU can read and write this register at any time. The SCI can modify RWU in some circumstances. In general, interrupts enabled by these control bits are cleared by reading SCSR, then reading (receiver status bits) or writing (transmitter status bits) SCDR.

Bit 15 — Not Implemented

LOOPS — Loop Mode

- 0 = Normal SCI operation, no looping, feedback path disabled
- 1 = Test SCI operation, looping, feedback path enabled

LOOPS controls a feedback path on the data serial shifter. When loop mode is enabled, SCI transmitter output is fed back into the receive serial shifter. TXD is asserted (idle line). Both transmitter and receiver must be enabled before entering loop mode.

WOMS — Wired-OR Mode for SCI Pins

- 0 = If configured as an output, TXD is a normal CMOS output.
- 1 = If configured as an output, TXD is an open-drain output.

WOMS determines whether the TXD pin is an open-drain output or a normal CMOS output. This bit is used only when TXD is an output. If TXD is used as a general-purpose input pin, WOMS has no effect.

ILT — Idle-Line Detect Type

- 0 = Short idle-line detect (start count on first one)
- 1 = Long idle-line detect (start count on first one after stop bit(s))

PT — Parity Type

- 0 = Even parity
- 1 = Odd parity

When parity is enabled, PT determines whether parity is even or odd for both the receiver and the transmitter.

8 Summary of Changes

This is a partial revision. Most of the publication remains the same, but the following changes were made to improve it. Typographical errors that do not affect content are not annotated. This document has also been reformatted for use on the web.

Pages 2-3	New Ordering Information included.
Page 6	New block diagram drawn.
Page 7	New 132-pin assignment diagram drawn.
Page 8	New 144-pin assignment diagram drawn.
Page 9	New address map drawn.
Pages 10-14	Added Signal Description section.
Pages 15-47	Expanded and revised SIM section. Made all register diagrams and bit mnemonics consistent. Incorporated new information concerning the system clock, resets, interrupts, and chip-selects circuits.
Page 48-56	Expanded and revised CPU section. Made all register diagrams and bit mnemonics consistent. Revised instruction set summary information.
Page 57-70	Expanded and revised TPU section. Made all register diagrams and bit mnemonics consistent. Revised time functions information to include both MC68332A and MC68332G microcode ROM applications.
Page 71-92	Expanded and revised QSM section. Made all register diagrams and bit mnemonics consistent. Added information concerning SPI and SCI operation.
Page 93-95	Revised Standby RAM with TPU Emulation RAM section. Made all register diagrams and bit mnemonics consistent.